FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC2A10 **COMPUTER SCIENCE 2A**
CAMPUS	AUCKLAND PARK CAMPUS (APK)
EXAM	JULY

DATE: 2016-07 **SESSION:** Main

LECTURER(S): **MR. A. MAGANLAL**
MR. B. GREAVES

MODERATOR: **DR. DT. VAN DER HAAR**

DURATION: 120 MINUTES **MARKS:** 100

Please read the following instructions carefully:

1. Answer **all** the questions.
2. Answer only in the examination books provided.
3. The use of calculators is *not* permitted.
4. Write *cleanly* and *legibly*.
5. This paper contains 10 questions.
6. This paper consists of 3 pages.

QUESTION 1

Java Overview

(a) **Discuss** the differences between *low-level* and *high-level* programming language categories. [04]

(b) **Name** and **discuss** two *features* of the `Java` language. [04]

(c) **Discuss** the differences between `C++` and `Java` with regards to *object allocation*. [02]

Total: 10

QUESTION 2

Elementary Java Programming

(a) **Provide** the storage size of the following *primitive data types*: **long**, **int**, **short**. [03]

(b) **How** can the *input* and *output* streams be referenced in a Java application? [02]

(c) **Classify** the *Selection Sort* algorithm in terms of: [05]

- Computational Complexity
- Memory usage
- Approach used
- Stability
- Method employed

Total: 10

QUESTION 3

Text Processing and Persistence

(a) **Provide** a *regular expression* that matches a list of heights with names. [05]

For example:

```
173 cm Mark
160 cm Thabo
152 cm Sue
```

(b) **Can** a **Scanner** be used to read binary data? **Provide** a reason for your answer. [03]

(c) When writing object using an **`ObjectOutpoutStream`** some data members cannot be written. **Which** data members cannot be serialized? [02]

Total: 10

QUESTION 4

Object Orientation

(a) **How** does a Java programmer prevent a method from being overridden? [02]

(b) **List** 3 *requirements* a class must conform to in order for it to be an *immutable class.* [04]

(c) Provide a **definition** of a *marker interface*. [02]

(d) **Identify** the *problem* with the following code. [02]

```
1 class Vehicle{}
2 class MotorCar extends Vehicle{}
3 public class Test {
4   public static void main(String[] args) {
5     Object instance = new Vehicle();
6     MotorCar gti = (MotorCar) instance;
7   }
8 }
```

Total: 10

QUESTION 5

Graphical User Interfaces

(a) **Name** and **describe** the three *Java GUI frameworks*. [09]

(b) **Name** one *layout manager* found in Java. [01]

Total: 10

QUESTION 6

Advanced Java Programming

(a) With regards to Java multi-threaded programming:

i. **Define** a *task*. [02]

ii. **Define** a *thread*. [02]

iii. **Discuss** how these two concepts are related. [01]

(b) **Define** *type erasure*. [03]

(c) **Name** two **methods** of a Java applet which are called by a browser. [02]

Total: 10

QUESTION 7

Design Patterns

(a) **Name** three of the *golden rules of design patterns*. [03]

(b) **Discuss** the limitations of the *Abstract Factory design pattern*. [04]

(c) **Name** three *structural design patterns*. [03]

Total: 10

QUESTION 8

UML

Provide a UML class diagram that illustrates the *Visitor design pattern*.

Total: 10

QUESTION 9

Cold Code

Provide Java source code for a ***paintComponent*** method which will draw a filled orange circle of 30 pixels at the center of the component.

Total: 10

QUESTION 10

Fill-in code

Read the code below and provide the missing code (in the segments labelled as A to G).

```
1  /* Imports omitted */
2  public class BinaryIO {
3    public void readApp(String path) {
4      File binFile = new File(path); // Get file handle
5      DataInputStream __(A (2 marks))__ = null;
6      try {
7        // Create required streams
8        FileInputStream fos = new FileInputStream(__(B (1 marks))__)
9        binin = new DataInputStream(new __(C (1 marks))__(fos));
10       // Read required data
11       String ID = binin.readUTF();
12       int mark = binin.readInt();
13       double rate = binin.__(D (2 marks))__();
14       System.out.printf("%s %d %f", ID, mark, rate); // Finish
             reading, display.
15     }
16     catch (__(E (1 marks))__ ex) {
17       ex.printStackTrace();
18     }
19     catch (IOException e) { e.printStackTrace(); }
20     finally {
21       if (binin != null) {
22         __(F (1 marks))__ {
23           binin.__(G (1 marks))__();
24         }
25         catch (IOException ex) { ex.printStackTrace(); }
26       }
27     }
28   }
29 }
```

Total: 10

The End!